Apache Hadoop

*Devadharshini Subiksha R
**Rathish S
***Mohana priya S

Abstract

The Hadoop Distributed File System (HDFS) is designed to store very large data sets reliably, and to stream those data sets at high bandwidth to user applications. In a large cluster, thousands of servers both host directly attached storage and execute user application tasks. By distributing storage and computation across many servers, the resource can grow with demand while remaining economical at every size.

1. INTRODUCTION

Hadoop is the solution to above Big Data problems. It is an open source software framework used to develop data processing applications which are executed in a distributed computing environment. Applications built using HADOOP are run on large data sets distributed across clusters of commodity computers. Commodity computers are cheap and widely available. These are mainly useful for achieving greater computational power at low cost.

It is developed as a project by Apache Software Foundation. Doug Cutting created Hadoop. In the year 2008 Yahoo gave Hadoop to Apache Software Foundation. Since then two versions of Hadoop has come. Version 1.0 in the year 2011 and version 2.0.6 in the year 2013. Hadoop comes in various flavors like Cloudera, IBM Big Insight, MapR and Hortonworks.

Similar to data residing in a local file system of a personal computer system, in Hadoop, data resides in a distributed file system which is called as a Hadoop Distributed File system. The processing model is based on 'Data Locality' concept wherein computational logic is sent to cluster nodes(server) containing data. This computational logic is nothing, but a compiled version of a program written in a high-level language such as Java. Such a program, processes data stored in Hadoop HDFS.

The shortcomings of the traditional approach which led to the invention of Hadoop –

1.1. Storage for Large Datasets:

The conventional RDBMS is incapable of storing huge amounts of Data. The cost of data storage in available RDBMS is very high. As it incurs the cost of hardware and software both.

1.2. Handling data in different formats:

The RDBMS is capable of storing and manipulating data in a structured format. But in the real world we have to deal with data in a structured, unstructured and semi-structured format.

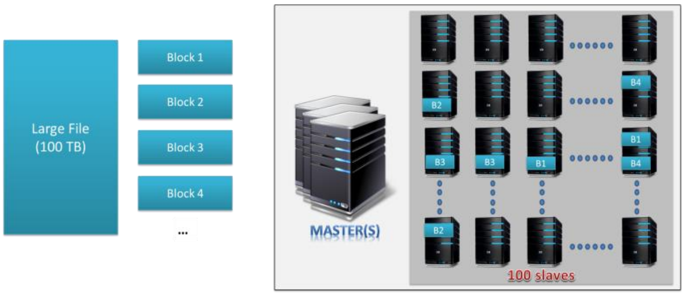
1.3. Data getting generated with high speed:

The data in oozing out in the order of tera to peta bytes daily. Hence we need a system to process data in real-time within a few seconds. The traditional RDBMS fail to provide real-time processing at great speeds.

2. CORE COMPONENTS OF HADOOP

2.1. HDFS:

Short for Hadoop Distributed File System provides for distributed storage for Hadoop. HDFS has a master-slave topology.



Master is a high-end machine where as slaves are inexpensive computers. The Big Data files get divided into the number of blocks. Hadoop stores these blocks in a distributed fashion on the cluster of slave nodes. On the master, we have metadata stored.

Name Node: Name Node performs following functions –

- Name Node Daemon runs on the master machine.
- It is responsible for maintaining, monitoring and managing Data Nodes.
- It records the metadata of the files like the location of blocks, file size, permission, hierarchy etc.
- Namenode captures all the changes to the metadata like deletion, creation and renaming of the file in edit logs.
- It regularly receives heartbeat and block reports from the DataNodes.

DataNode: The various functions of DataNode are as follows –

- DataNode runs on the slave machine.

- It stores the actual business data.
- It serves the read-write request from the user.
- DataNode does the ground work of creating, replicating and deleting the blocks on the command of NameNode.
- After every 3 seconds, by default, it sends heartbeat to NameNode reporting the health of HDFS.

2.2. MapReduce:

It is the data processing layer of Hadoop. It processes data in two phases.

They are:-

Map Phase- This phase applies business logic to the data. The input data gets converted into key-value pairs.

Reduce Phase- The Reduce phase takes as input the output of Map Phase. It applies aggregation based on the key of the key-value pairs.

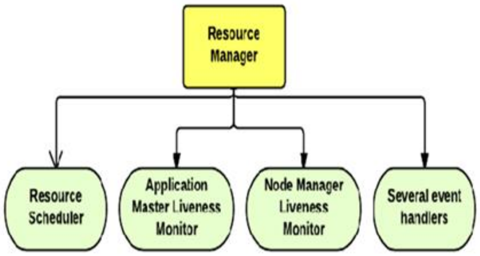
Map-Reduce works in the following way:

- The client specifies the file for input to the Map function. It splits it into tuples
- Map function defines key and value from the input file. The output of the map function is this key-value pair.
- MapReduce framework sorts the key-value pair from map function.
- The framework merges the tuples having the same key together.
- The reducers get these merged key-value pairs as input.
- Reducer applies aggregate functions on key-value pair.
- The output from the reducer gets written to HDFS.

2.3 YARN

Short for Yet another Resource Locator has the following components:-

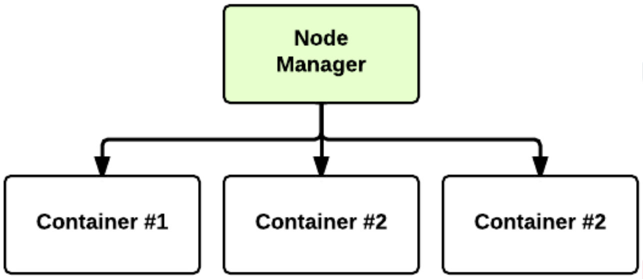
Resource Manager


Resource Manager
Resource Scheduler
Application Master Liveness Monitor
Node Manager Liveness Monitor
Several event handlers

- Resource Manager runs on the master node.
- It knows where the location of slaves (Rack Awareness).
- It is aware about how much resources each slave have.
- Resource Scheduler is one of the important service run by the Resource Manager.
- Resource Scheduler decides how the resources get assigned to various tasks.

- Application Manager is one more service run by Resource Manager.
- Application Manager negotiates the first container for an application.
- Resource Manager keeps track of the heart beats from the Node Manager.

Node Manager



- It runs on slave machines.
- It manages containers. Containers are nothing but a fraction of Node Manager’s resource capacity
- Node manager monitors resource utilization of each container.
- It sends heartbeat to Resource Manager.

3. FEATURES OF HADOOP

Apache Hadoop is the most popular and powerful big data tool, Hadoop provides world’s most reliable storage layer – HDFS, a batch Processing engine – MapReduce and a Resource Management Layer – YARN. In this section of features of Hadoop, Let us discuss important features of Hadoop which are given below-

3.1. Open Source

Apache Hadoop is an open source project. It means its code can be modified according to business requirements.

3.2. Distributed Processing

As data is stored in a distributed manner in HDFS across the cluster, data is processed in parallel on a cluster of nodes.

3.3. Fault Tolerance

This is one of the very important features of Hadoop. By default 3 replicas of each block is stored across the cluster in Hadoop and it can be changed also as per the requirement. So if any node goes down, data on that node can be recovered from other nodes easily with the help of this characteristic. Failures of nodes or tasks are recovered automatically by the framework. This is how Hadoop is fault tolerant.

3.4. Reliability

Due to replication of data in the cluster, data is reliably stored on the cluster of machine despite machine failures. If your machine goes down, then also your data will be stored reliably due to this characteristic of Hadoop.

3.5. High Availability

Data is highly available and accessible despite hardware failure due to multiple copies of data. If a machine or few hardware crashes, then data will be accessed from another path.

3.6. Scalability

Hadoop is highly scalable in the way new hardware can be easily added to the nodes. This feature of Hadoop also provides horizontal scalability which means new nodes can be added on the fly without any downtime.

3.7. Economic

Apache Hadoop is not very expensive as it runs on a cluster of commodity hardware. We do not need any specialized machine for it. Hadoop also provides huge cost saving also as it is very easy to add more nodes on the fly here. So if requirement increases, then you can increase nodes as well without any downtime and without requiring much of pre-planning.

3.8. Easy to use

No need of client to deal with distributed computing, the framework takes care of all the things. So this feature of Hadoop is easy to use.

3.9. Data Locality

This one is a unique features of Hadoop that made it easily handle the Big Data. Hadoop works on data locality principle which states that move computation to data instead of data to computation. When a client submits the MapReduce algorithm, this algorithm is moved to data in the cluster rather than bringing data to the location where the algorithm is submitted and then processing it.

4.PROS AND CONS OF HADOOP

Hadoop is easy to use, scalable and cost-effective. Along with this, Hadoop has many advantages. So, following are the pros of Hadoop that makes it so popular –

4.1 Varied Data Sources

Hadoop accepts a variety of data. Data can come from a range of sources like email conversation, social media etc. and can be of structured or unstructured form. Hadoop can derive value from diverse data. Hadoop can accept data in a text file, XML file, images, CSV files etc.

4.2. Cost-effective

Hadoop is an economical solution as it uses a cluster of commodity hardware to store data. Commodity hardware is cheap machines hence the cost of adding nodes to the framework is not much high. In Hadoop 3.0 we have only 50% of storage overhead as opposed to 200% in Hadoop2.x. This requires less machine to store data as the redundant data decreased significantly.

4.3. Performance

Hadoop with its distributed processing and distributed storage architecture processes huge amounts of data with high speed. Hadoop even defeated supercomputer the fastest machine in 2008. It divides the input data file into a number of blocks and stores data in these blocks over several nodes. It also divides the task that user submits into various sub-tasks which assign to these worker nodes containing required data and these sub-task run in parallel thereby improving the performance.

4.4. Fault-Tolerant

In Hadoop 3.0 fault tolerance is provided by erasure coding. For example, 6 data blocks produce 3 parity blocks by using erasure coding technique, so HDFS stores a total of these 9 blocks. In event of failure of any node the data block affected can be recovered by using these parity blocks and the remaining data blocks.

4.5. Highly Available

In Hadoop 2.x, HDFS architecture has a single active NameNode and a single Standby NameNode, so if a NameNode goes down then we have standby NameNode to count on. But Hadoop 3.0 supports multiple standby NameNode making the system even more highly available as it can continue functioning in case if two or more NameNodes crashes.

4.6. Low Network Traffic

In Hadoop, each job submitted by the user is split into a number of independent sub-tasks and these sub-tasks are assigned to the data nodes thereby moving a small amount of code to data rather than moving huge data to code which leads to low network traffic.

4.7. High Throughput

Throughput means job done per unit time. Hadoop stores data in a distributed fashion which allows using distributed processing with ease. A given job gets divided into small jobs which work on chunks of data in parallel thereby giving high throughput.

4.8. Open Source

Hadoop is an open source technology i.e. its source code is freely available. We can modify the source code to suit a specific requirement.

4.9. Scalable

Hadoop works on the principle of horizontal scalability i.e. we need to add the entire machine to the cluster of nodes and not change the configuration of a machine like adding RAM, disk and so on which is known as vertical scalability. Nodes can be added to Hadoop cluster on the fly making it a scalable framework.

4.10. Ease of use

The Hadoop framework takes care of parallel processing, MapReduce programmers does not need to care for achieving distributed processing, it is done at the backend automatically.

4.11. Compatibility

Most of the emerging technology of Big Data is compatible with Hadoop like Spark, Flink etc. They have got processing engines which work over Hadoop as a backend i.e. We use Hadoop as data storage platforms for them.

4.12. Multiple Languages Supported

Developers can code using many languages on Hadoop like C, C++, Perl, Python, Ruby, and Groovy.

Cons of Hadoop.

4.13. Issue With Small Files

Hadoop is suitable for a small number of large files but when it comes to the application which deals with a large number of small files, Hadoop fails here. A small file is nothing but a file which is significantly smaller than Hadoop's block size which can be either 128MB or 256MB by default. These large number of small files overload the Namenode as it stores namespace for the system and makes it difficult for Hadoop to function.

4.14. Vulnerable By Nature

Hadoop is written in Java which is a widely used programming language hence it is easily exploited by cyber criminals which makes Hadoop vulnerable to security breaches.

4.15. Processing Overhead

In Hadoop, the data is read from the disk and written to the disk which makes read/write operations very expensive when we are dealing with tera and petabytes of data. Hadoop cannot do in-memory calculations hence it incurs processing overhead.

4.16. Supports Only Batch Processing

At the core, Hadoop has a batch processing engine which is not efficient in stream processing. It cannot produce output in real-time with low latency. It only works on data which we collect and store in a file in advance before processing.

4.17. Iterative Processing

Hadoop cannot do iterative processing by itself. Machine learning or iterative processing has a cyclic data flow whereas Hadoop has data flowing in a chain of stages where output on one stage becomes the input of another stage.

4.18. Security

For security, Hadoop uses Kerberos authentication which is hard to manage. It is missing encryption at storage and network levels which are a major point of concern.

5. CONCLUSION

The thesis has given a brief introduction to the core technology of Hadoop but there are still many applications and projects developed on Hadoop. In conclusion, the Hadoop, which is based on the Hadoop HDFS and Map Reduce has provided a distributed data processing platform. The high fault tolerance and high scalability allow its users to apply Hadoop on cheap hardware. The MapReduce distributed programming mode allows the users to develop their own applications without the users having to know the bottom layer of the MapReduce. Because of the advantages of Hadoop, the users can easily manage the computer resources and build their own distributed data processing platform. Above all, it is obvious to notice the convenience that the Hadoop has brought in Big Data processing. It also should be pointed out that since Google published the first paper on the distributed file system till now, the history of Hadoop is only 10-year old. With the advancement of the computer science and the Internet technology, Hadoop has rapidly solved key problems and been widely used in real life. In spite of this, there are still some problems in facing the rapid changes and the ever increasing demand of analysis. To solve these problems, Internet companies, such as Google also introduced the newer technologies. It is predictable that with the key problems being solved, Big Data processing based on Hadoop will have a wider application prospect.

REFERENCE LINKS:

1. https://www.seminarsonly.com/computer%20science/Hadoop.php
2. https://www.guru99.com/learn-hadoop-in-10-minutes.html
3. https://studymafia.org/hadoop-seminar-ppt-with-pdf-report/
4. https://www.tutorialspoint.com/hadoop/hadoop_enviornment_setup.html